ISO/IEC JTC1/SC2/WG2 N4410
ISO/IEC JTC1/SC2/WG2/IRG N1843_ROK
Doc. #: Korea JTC1/SC2 K2165_1

Korea JTC1/SC2, Committee on Character Codes

Author: YANG Wangsung; KIM, Kyongsok
Date: 2012.11.08.
Status: National Body Position, Rep. of KOREA
Subject: comments RE: WG2 N4230 (=IRG N1843) UNC process

R.O.Korea reviewed WG2 N4230 (=IRG N1843), analyzed it and then suggests a few things.

WG2 N4230 (=IRG N1843), Proposal to establish a CJK Unified Ideographs "Urgently Needed Characters" process

1. Background: Portions extracted from WG2 N4230 (=IRG N1843)

...
The process of standardizing repertoires of CJK Unified Ideographs is long and cumbersome, and is almost always measured in years. This is primarily because the typical CJK Unified Ideograph repertoire includes thousands or tens of thousands of characters, ...

To address the size issue of the repertoire, in that it must be small, we recommend that each national body be limited to 25 character submissions per repertoire, ...

2. Analysis: Why did it take so much time in the past to finish one CJKU block

1) The size of a block was too big

- CJKU main = 20,952 chars, CJKU ExtB = 42,711 chars,
  (In contrast, CJKU ExtA and ExtC are not so big relatively: CJKU ExtA = 6,582 chars, CJKU ExtC = 4,149 chars)
- Some charactered were removed from the original submissions during review process in each block. Therefore the size of each block reviewed by IRG is even bigger.

==> Now IRG PnP limits the size of one CJKU block to 4,000 chars. Therefore it won't take so much time in the future as in the past.

2) The quality of submissions were not high.

- There were many errors/problems/etc since there was no penalty for low quality submissions.

==> Now IRG has the 5% rule. Therefore low-quality submission will be removed as a whole from the candidate list of chars for a new CJKU block.

A relevant portion is quoted below from IRG N1823 IRG PnP Draft3:

2.2.6. Quality Assurance: The 5% Rule

For any character encoding standard, a common general principle is to encode the same character once and only once. Before any submission, it is the submitter's responsibility to filter out the ideographs that are already in the ISO/IEC 10646 international coding standard:

- *the published standard,*
- *any of its published amendments,*
- *any of its amendments under ballot in JTC1/SC2, or*
- *one of the working sets of the IRG.*

In assessing the suitability of a proposed ideograph for encoding, the IRG will evaluate the credibility and quality of the submitter's proposal. If the IRG finds more than 5% of duplicated characters in the above mentioned collections from the submitter's source set during the IRG review process, the whole submission will be removed from the subsequent IRG working drafts for that particular IRG project.

3) New format tables of CJKU main, ExtA, and ExtB in ISO/IEC 10646, 2ed and 3ed, using TTF fonts in multi-column format were reviewed in parallel with standardizing ExtE.

- It explains why it took so much time to finalize ExtE.
- Therefore UNC was introduced (later became ExtD) and finalzied before ExtE which started earlier than ExtE(UNC).

==> This kind of review process of new format tables will not take place periodically or frequently in the future. ExtE review process was exceptional and we should be careful not to take it as a typical process and justify some other alternative process.

3. R.O.Korea's suggestion

- Since having too many small CJKU blocks is not so desirable, R.O.Korea suggests that UNC process be used "sparingly" only when it is "really" needed.

- R.O.Korea suggests that, even when we finalize a small set of UNC chars, whenever possible, we do not add a UNC as a separate CJKU block to UCS; instead, we append the UNC at the end of the current working set, say ExtX, so that we do not have a small CJKU block containing only a small number of UNC chars.

- This is explained pictorially below:

a) two working sets are reviewed by IRG in parallel:

```
| originalExtX                    |      | UNC  |
```

b) Suppose that, when UNC is finalized, originalExtX is also being finalized. In such a situation, we can handle in three different ways.
- R.O.Korea recommends method 3.

b-1) method 1: two separate CJKU blocks -- ExtX and ExtY block

```
<---- ExtX -------------------->      <-ExtY->
| originalExtX                  |      | UNC  |
```

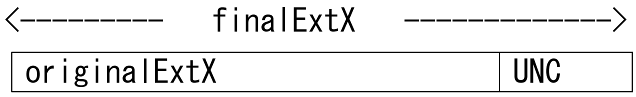
b-2) method 2: one final CJKU ExtX block with original ExtX and UNC merged and reordered

- All the characters in the original ExtX and UNC blocks are merged and reordered according to radical-stroke order.
- Although we do not have small CJKU blocks, it is time-consuming and error-prone to merge and reorder them and therefore it is not a desirable method.

```
<---------   finalExtX   ------------->
| originalExtX + UNC                    |
```

b-3) method 3: one final CJKU ExtX block with UNC appended at orig. ExtX

- Characters in the original ExtX and UNC blocks are NOT merged or reordered according to radical-stroke order.
- We just append UNC at the end of a original ExtX and name the new block ExtX as a whole.
- There is no time-consuming process while we avoid having small CJKU blocks.
- It is a desirable method.
- R.O.Korea recommends this method (i.e., method 3).



＊＊＊